# Introduction of Raspberry Pi Pico

## 1. Raspberry Pi Pico

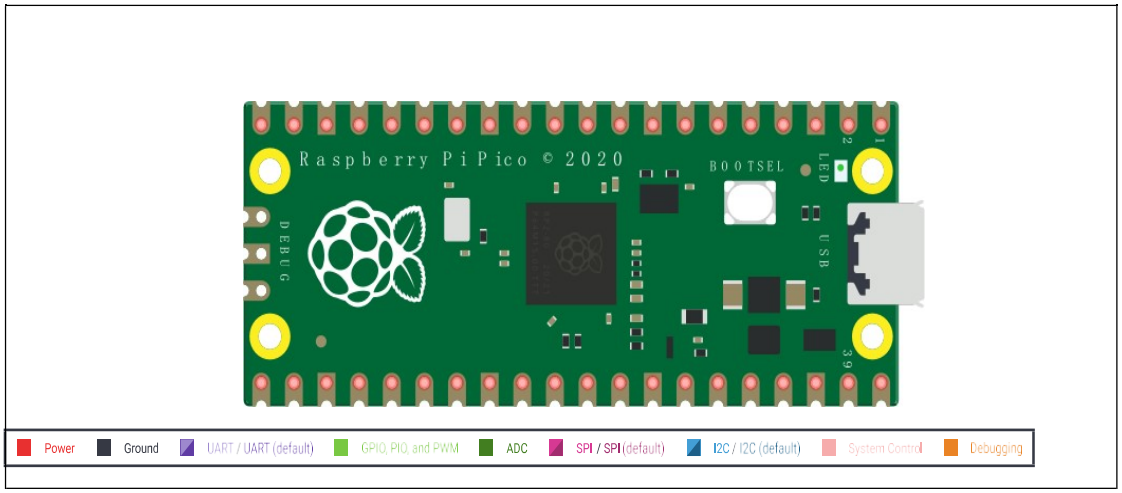
### 1.1 Technical Specification

Raspberry Pi Pico is a low-cost, high-performance microcontroller board with flexible digital interfaces. Key features include:

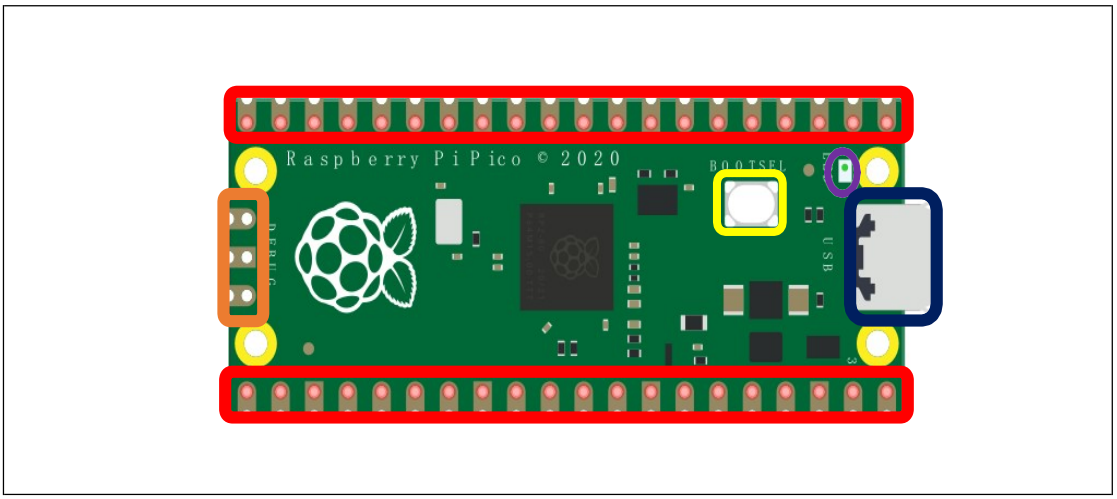
- ✧ RP2040 microcontroller chip designed by Raspberry Pi in the United Kingdom
- ✧ Dual-core Arm Cortex M0+ processor, flexible clock running up to 133 MHz
- ✧ 264KB of SRAM, and 2MB of on-board Flash memory
- ✧ Castellated module allows soldering direct to carrier boards
- ✧ USB 1.1 with device and host support
- ✧ Low-power sleep and dormant modes
- ✧ Drag-and-drop programming using mass storage over USB
- ✧ 26 × multi-function GPIO pins
- ✧ 2 × SPI, 2 × I2C, 2 × UART, 3 × 12-bit ADC, 16 × controllable PWM channels
- ✧ Accurate clock and timer on-chip
- ✧ Temperature sensor
- ✧ Accelerated floating-point libraries on-chip
- ✧ 8 × Programmable I/O (PIO) state machines for custom peripheral support

### 1.1.1 Pinout and Design Files

Raspberry Pi Pico is a light-weight electronic product with tiny size and low price. From the picture below we can see that its onboard resources have been connected to the edge interface, which is very suitable for electronic enthusiasts to use in DIY.

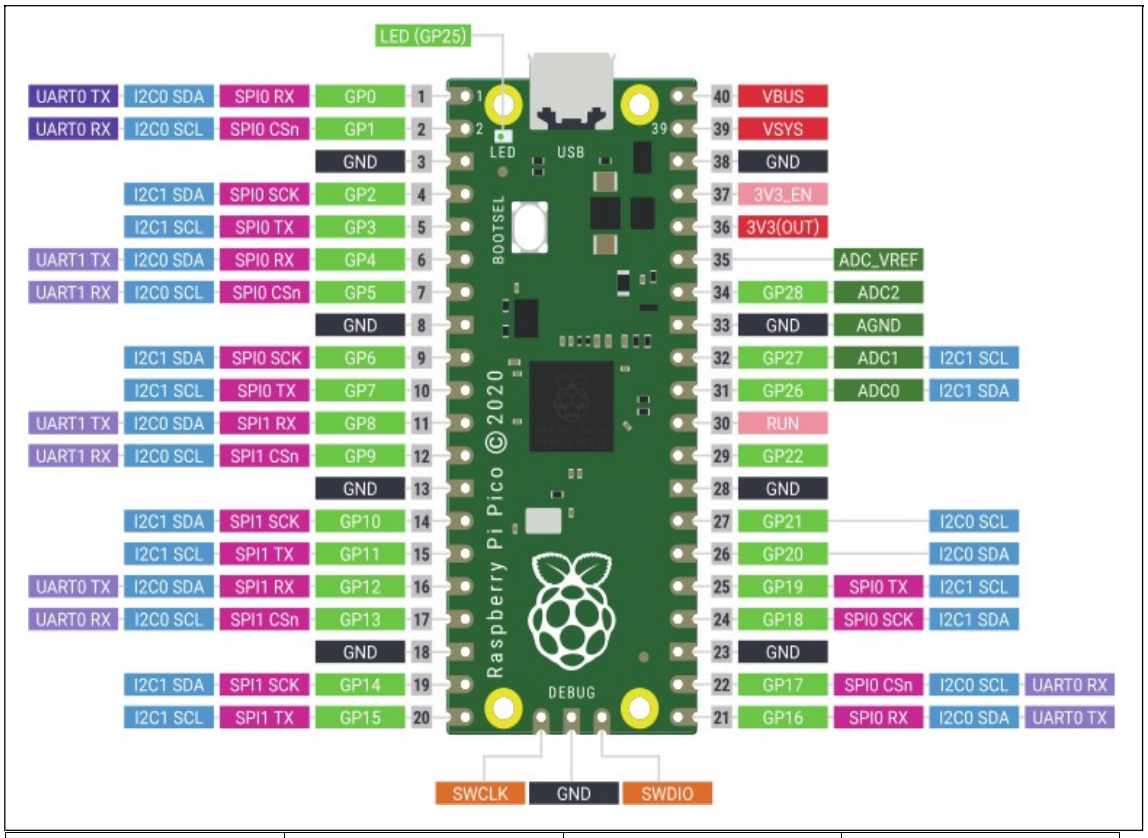



The hardware interfaces are distributed as follows:




| Frame color | Description |
|---|---|
| | Pins |
| | BOOTSE button |
| | USB port |
| | LED |
| | Debugging |

Function definition of pins:




| Color | Pins | Color | Pins |
|---|---|---|---|
| | GND | | Power |
| | GPIO | | ADC |
| | UART(defua lt) | | UART |
| | SPI | | I2C |
| | System Control | | Debugg ing |

For details: https://datasheets.raspberrypi.org/pico/pico-datasheet.pdf

| | |
|---|---|
| GND | Ground Pin |
| Power | VBUS(microUSB Voltage)、VSYS(2-5VDC Input)、3V3(3.3V OUT)、3V3_EN(Enables Pico) |
| System Control | Run(Start or disable RP2040 microcontroller or reset) |
| ADC | Raspberry Pi Pico has a total of 5 ADC with a resolution of 12 bits, which are ADC0(GP26)，ADC1(GP27)，ADC2(GP28)，ADC3(GP29), ADC4 respectively. Among<br>them, ADC3(GP29) is used to measure the VSYS on Pico board; ADC4 is derectly connected to the RP2040's built-in tempertature sensor.<br>ADC_VREF can connect to external accurate voltmeter as ADC reference.<br>ADC_GND pin is used as the reference<br>point for grounding. |
| PWM | There are 16 PWM channels on Raspberry Pi Pico, each of which can control frequency<br>and duty cycle independently. The GPIO pins are switched to PWM function. |
| UART | There are 2 UART: UART0，UART1. |
| SPI | There are 2 SPI: SPI0，SPI1. |
| I2C | 2 I2C: I2C0，I2C1. |
| Debugging | It is used when debugging code. |

# UART, I2C, SPI Defalt Pin

## UART

| Function | Default |
|---|---|
| UART_BAUDRATE | 115200 |
| UART_BITS | 8 |
| UART_STOP | 1 |
| UART0_TX | Pin 0 |
| UART0_RX | Pin 1 |
| UART1_TX | Pin 4 |
| UART1_RX | Pin 5 |

## I2C

| Function | Default |
|---|---|
| I2C Frequency | 400000 |
| I2C0 SCL | Pin 9 |
| I2C0 SDA | Pin 8 |
| I2C1 SCL | Pin 7 |
| I2C1 SDA | Pin 6 |

## SPI

| Function | Default |
|---|---|
| SPI_BAUDRATE | 1000000 |
| SPI_POLARITY | 0 |
| SPI_PHASE | 0 |
| SPI_BITS | 8 |
| SPI_FIRSTBIT | MSB |
| SPI0_SCK | Pin 6 |
| SPI0_MOSI | Pin 7 |
| SPI0_MISO | Pin 4 |
| SPI1_SCK | Pin 10 |
| SPI1_MOSI | Pin 11 |
| SPI1_MISO | Pin 8 |

For more detailed information, please refer to:
https://datasheets.raspberrypi.org/pico/raspberry-pi-pico-python-sdk.pdf

Download the Pinout Diagram (PDF)

## Design Files

Download Design Files (Cadence Allegro)
Download STEP File
Download Fritzing Part

**NOTE**: More information on Fritzing is available on the fritzing.org web site.

## Documentation

Documentation for Raspberry Pi Pico and other RP2040-based boards.

## RP2040 Device

RP2040 Datasheet
A microcontroller by Raspberry Pi Hardware design with RP2040
Using RP2040 microcontrollers to build boards and products

### 1.1.2 Raspberry Pi Pico

Raspberry Pi Pico Datasheet

An RP2040-based microcontroller board.

Getting started with Raspberry Pi Pico

C/C++ development with Raspberry Pi Pico and other RP2040-based microcontroller boards

**NOTE:** While it is not officially supported there is a Pico Setup for Windows installation tool which automates installation of the C/C++ SDK on Windows 10.

### 1.1.3 Software Development

Raspberry Pi Pico C/C++ SDK

Libraries and tools for C/C++ development on RP2040 microcontrollers

Raspberry Pi Pico Python SDK

A MicroPython environment for RP2040 microcontrollers

The API level Doxygen documentation for the Raspberry Pi Pico C/C++ SDK is also available as a micro-site.

**NOTE:** If you are building applications with the C/C++ SDK and targeting boards other than the Raspberry Pi Pico, you will need to pass -DPICO_BOARD=boardname to CMake. Here boardname is the name of your board, e.g. for the Adafruit Feather RP2040 you should pass -DPICO_BOARD=adafruit_feather_rp2040. See the boards/ directory in the Pico SDK, and the forums, for more information.

## 1.2 Software Utilities

### 1.2.1 What is on your Pico?

If you have forgotten what has been programmed into your Raspberry Pi Pico, and the program was built using our Pico C/C++ SDK, it will usually have a name and other useful information embedded into the binary. You can use the Picotool command line utility to find out these details. Full instructions on how to use Picotool to do this are available in our 'getting started' documentation.

- Go to the Picotool Github repository.

### 1.2.2 Debugging using another Raspberry Pi Pico

It is possible to use one Raspberry Pi Pico to debug another Pico. This is possible via picoprobe, an application that allows a Pico to act as a USB → SWD and UART converter. This makes it easy to use a Pico on non-Raspberry Pi platforms such as Windows, Mac, and Linux computers where you don't have GPIOs to connect directly to your Pico. Full instructions on how to use Picoprobe to do this are available in our 'getting started' documentation.

- Download the UF2 file
- Go to the Picoprobe Github repository

### 1.2.3 Resetting Flash memory

Pico's BOOTSEL mode lives in read-only memory inside the RP2040 chip, and can't be overwritten accidentally. No matter what, if you hold down the BOOTSEL button when you plug in your Pico, it will appear as a drive onto which you can drag a new UF2 file. There is no way to brick the board through software. However, there are some circumstances where you might want to make sure your Flash memory is empty. You can do this by dragging and dropping a special UF2 binary onto your Pico when it is in mass storage mode.

- Download the UF2 file
- See the code on Github

## 1.3 Reference Material

Raspberry Pi Pico chinese website:

https://pico.org.cn/

Documentation of RasberryPi :

https://www.raspberrypi.org/documentation/pico/getting-started/

Raspberry Pi datasheet:

https://datasheets.raspberrypi.org/pico/pico-datasheet.pdf

RP2040 datashee:

https://datasheets.raspberrypi.org/rp2040/rp2040-datasheet.pdf

RP2040 hardware design:

https://datasheets.raspberrypi.org/rp2040/hardware-design-with-rp2040.pdf

Getting stared withPico:

https://datasheets.raspberrypi.org/pico/getting_started_with_pico.pdf

Pico C/C++ software development kit:

https://datasheets.raspberrypi.org/pico/sdk/pico_c_sdk.pdf

Pico Python software development kit:

https://datasheets.raspberrypi.org/pico/sdk/pico_python_sdk.pdf

## 1.4 What's Next?

THANK YOU for reading this document of this Pico board !

If you find errors, omissions or you have suggestions and/or questions about this document, please feel free to contact us: cokoino@outlook.com

We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website: http://cokoino.com/ We will continue to launch fun, cost-effective, innovative and exciting products.

Thank you again for choosing Cokoino products.